IBM SOA Customer Experience Sharing
- Operationalizing SOA Business Transformation

Jenny Choy
Distinguished Engineer
IBM Global Services
jnchoy@ca.ibm.com
1-905-316-2247
Agenda

- The Business Motivation of SOA

- Best Practices for Operationalizing SOA Business Transformation
  1. Take on a Program Approach for your SOA Implementation
  2. Use a Business Architecture to Provide Guidance for Technical Solution Designs
  3. Build an SOA Transformation Technology Architecture
  4. Create Pragmatic Roadmap and Demonstrate Continuous Progress
  5. Cross Project Service Reuse planning
  6. Pay Attention to implementing products and tools for Common Services
  7. Special Care with Coexistence and Migration of Existing Applications
Our clients face a complex array of challenges

**Increase revenue**
- Increase customer profitability
- Create new routes to market
- Create new value from existing systems

**Integrate across the enterprise**
- Integrate historically separate systems
- Facilitate mergers and acquisitions of enterprises
- Synchronize common information across all systems

**Drive down cost**
- Eliminate duplicate systems
- Improve IT and operations efficiency
- Reuse current functions
- Build once and leverage
- Improve time to market

**Provide a flexible business model**
- Respond to market changes more quickly

**Reduce cycle times and cost for external business partners**
- Move from manual to automated transactions
- Facilitate flexible dealings with business partners

**Reduce risks and exposure**
- Improve visibility into business operations
- Regulatory compliance
- Align IT Processes and systems management
- Improve Business resilience
- Security

**SOA Transformation Requires Tightly Integrated Business and IT**
Leading companies are adopting a more componentized approach, realizing substantial changes

The evolution of business structure – tomorrow

CEO Needs
- Revenue growth with cost containment
- Key competency: responsiveness
- Critical success factor: enable effectiveness of people and processes

CIO Challenges
- Aligning IT and business goals to grow revenue and contain costs
- Building responsiveness and agility into the organization through IT
- How can IT help enable people and teams to be more effective

Source: CEO Study of 456 WW CEOs, IBM Corp. 2004
Source: Operating Environment Market Drivers Study, IBM Corp. 2004

- Dynamically configurable architecture with business service provisioning
The Business Motivation of Service Oriented Architecture (SOA)

- Improve Business agility
  - Faster Delivery of New Products Functions and Services

- Flexible business infrastructure
  - Ability to support Mergers / Acquisition
  - Allow Incremental Transformation without interrupting business

- Enable Streamlined Processes
  - Continuous Business Process Improvement
  - Business Process visibility
  - Better, Risk-Mitigated Decisioning
  - Reduced Costs
  - Improve financial reporting and business operations

- Higher Customer Satisfaction
  - Support multiple devices in multiple business channels using same application base
  - Better customer service
  - Buy application components (e.g. Rules Engines, Rating Engines) vs. build
  - Ability to choose vendors and suppliers

- Better IT Services
  - Allow for more consistent, sharable Infrastructure through standards and shared services
  - Reduce overall development, integration and maintenance complexity and costs
  - Faster Functional Development

- Leverage Existing Assets and Asset Reuse
  - Reuse of business services, application services and technical infrastructure services
The Concepts of Componentization and Service Orientation support dynamically configurable business services

By decomposing business needs into necessary services and composing combinations of existing and newly created services that represent business processes …

1. Decompose the Business Need into its necessary function and services (Separation of Concerns)
2. Reuse/Create Business Application Specific function and services (Create/reuse services)
3. Utilize common services provided by the Operating Environment (Leverage Infrastructure)

RESULT: Implemented Composite Application representing a business process
The Reality – Transformation speed is hindered by existing Business, Application and Infrastructure environments

**Legacy Spaghetti**

- Complexity of enterprise application portfolio
- Disjointed business processes
- Disjointed integration approaches
- Limited and diminishing skills in core applications
- Embedded processes and rules
- Old and inflexible IT systems that still run
- High cost of running the existing IT infrastructure
- Different Development methodologies

The concepts of a Service Oriented Architecture (SOA) will require a corresponding On Demand Operating Environment (ODOE)

Each element of the architecture is a service that together supports the SOA Transformation
Agenda

- The Business Motivation of SOA
- Best Practices for Operationalizing SOA Business Transformation

1. Take on a Program Approach for your SOA Implementation

2. Use a Business Architecture to Provide Guidance for Technical Solution Designs
3. Build a Prescriptive SOA Transformation Technology Architecture
4. Create Pragmatic Roadmap and Demonstrate Continuous Progress
5. Cross Project Service Reuse planning
6. Pay Attention to implementing products and tools for Common Services
7. Special Care with Coexistence and Migration of Existing Applications
Business-focused Implementation of SOA will put the output from SOMA into an integrated project (or groups of projects) context.
Business-Focused SOA Implementation Programs may involve simultaneous execution of one or more project elements.
Enterprise SOA Business Case & ROI

- Large scale programs for SOA require more planning and justification
  - May appear to take longer, Higher Risks, Higher Costs
  - Need an incremental transformation strategy

- Higher Benefits of SOA occur at the enterprise level
  - Small, web services projects provides limited business gains
  - Large scale implementations will drive strategic, high value impact for the enterprise
  - Need Governing Program level Architecture to govern technical investments

- Need to establish metrics for other tangible and intangible gains
  - Difficult to justify business case if only on cost cutting
  - Must also plan for gains from Revenue Generation functions such as multi-channel enablement
    - Process improvements and reduction of required support (e.g. less calls at helpdesk)
    - Reused services
    - Development efficiency and downstream maintenance reduction
    - Platform simplifications

- Need to establish project level process and governance for metrics and estimating models (both forecast & actual) for on going transformation
  - Incremental projects must have a quantifiable business benefit, and within each SOA Transformation step
  - Projects should roll up to program level

- Initial budget must include one or more pilot applications
  - Do not forget foundational shared services projects costs and spread
  - Cannot burden all SOA infrastructure set up costs to one single project
  - Some overhead must be included to cover learning curve, reusability assessment & design, addition of new infrastructure, tools, etc.

- Don’t forget Costs of integration costs and Costs for Co-existence
Scope Analysis is required in early days (and continuously) to ascertain where the greatest value can be achieved.

**Greatest Value**
- Focusing on the areas which drive value and are most likely to be successful
- Need to look across the businesses and consider all transformational elements (i.e., imaging, document management, STP, workflow, etc.)
- Area of diminishing returns may still be attractive

**Lesser Value I**
- Scope too narrow to drive large $ savings
- Enterprise infrastructure too expensive for in-scope operations
- If scope can’t be expanded, best tackled through incremental and tactical change

**Lesser Value II**
- Scope too wide to get organizational traction
- Complexity of requirements and business structure overwhelms program
- Knowing where to ‘cut-off’ depth of enterprise infrastructure is as much ‘art as science’
A Program View for SOA Transformation will provide a consistent view of Project Implementation, balancing Value, Time & Costs.
Program Management and Governance is key to success

- Overall Program Vision need to continuously govern the transformation initiatives
  - Must manage to Business Case Critical Success Factors

- Program Management
  - Multiple project streams running in parallel will require tighter coordination and management.
  - Require linkage of high level program management through to project management
  - Program communication is key
  - Each initiative need to consider impact on business from organization perspective

- Governance
  - Traceability and follow through of program framework and established guiding principle

- Business and IT need to collaborate as a program
  - Both business and IT need to understand why they are doing this.
  - Project teams must be aligned with overall SOA vision

- Program Infrastructure need to be established, with clear accountability, roles and responsibilities
  - Program (business stakeholders, executives, program management, project management)
  - business (business users, business analysts)
  - Technical team (architect, designers, developers)
  - Experienced larger scale program management skills required

- Iterative development – but make sure you have governance in place first
Implementing Multiple, Cross-functional SOA Projects require a disciplined approach

- **Pilot SOA Project(s)**
  - Must be small, but visible
  - Business Benefit delivery within 6-9 months with initial launch
  - Establish measurement & criteria

- **Establish standard templates and metrics for project business case and estimating methodology**
  - Top down Estimating will provide a quicker estimate for budgeting purposes
  - Bottom-up estimate will take too long
    - Do not be tempted to allow different development groups use own estimating methodology
    - “Not invented here” must be discouraged and fully justified
  - Include efforts for new service design, development, training for each project

- **Development Approach, Code Configuration, Tooling, Repository should be standardized**

- **Need to consider sufficient Infrastructure and Testing Environment across projects and releases**
  - Development
    - Development Integration Test – Key for multiple component development streams
    - Systems Integration Test
    - User Acceptance Test
    - Performance and Load,
    - Staging & Production environments
A Governance Framework should indicate an organization’s vision for centralized activities at the “enterprise” level, and decentralized activities, at the “Competency” level

1. **Strategic Governance** is responsible for strategy, funding and setting the direction of the program.
2. **Technology Governance** is responsible for technology aspects from both the enterprise and function perspectives.
3. **Services (COE) Governance** is responsible for the specific technology view and is generally
In order to achieve the desired business benefits, a holistic view needs to be taken. This means a single design authority, an end-to-end design method and overall program management.
Agenda

- The Business Motivation of SOA
- Best Practices for Operationalizing SOA Business Transformation
  1. Take on a Program Approach for your SOA Implementation
  2. Use a Business Architecture to Provide Guidance for Technical Solution Designs
  3. Build a Prescriptive SOA Transformation Technology Architecture
  4. Create Pragmatic Roadmap and Demonstrate Continuous Progress
  5. Cross Project Service Reuse planning
  6. Pay Attention to implementing products and tools for Common Services
  7. Special Care with Coexistence and Migration of Existing Applications
Successful SOA Implementation must consider all 3 Key Domains – Business, Application and Infrastructure

**Business Architecture Domain**
- Realign key parts of the business around the component model
- Provide a consistent vocabulary and vision to support transformation within and across domains

**Application Architecture Domain**
- Establish a set of reusable services that support rapid implementation of business functions
- Provide application functions that align with the business focus
- Provide a set of development tools to support the institution

**Technical Architecture and Operations Domain**
- Provide a flexible technical base for internal and external user access.
- Provide technical capabilities enabling integration of process elements
- Provide overall access to and control of the resources in the institution’s infrastructure
- Provide a flexible technical base supporting institution access to all data
- Provide flexible physical infrastructure throughout the institution
- Develop plans to address the incompleteness, inaccuracies and inconsistencies in the institution’s data

Insure that the information required across the organization is timely, accurate and consistent

Re-engineer high impact business processes to support the institution’s strategies

Provide a consistent vocabulary and vision to support transformation within and across domains

Realign key parts of the business around the component model

Provide overall access to and control of the resources in the institution’s infrastructure

Provide flexible physical infrastructure throughout the institution
There need to be a tight alignment between Strategy, Architectures and Implementation

**Strategy**

- "the city’s purpose & goals"

**Enterprise Architecture**

- "the city plan"

**System Design**

- "the buildings"

---

**IBM Global Services**

---
Transformation Projects must always have line of sight of the Business Architecture and Business Strategy

Why the focus on Business Architecture?

- A Business Architecture (BA)
  - Provides the framework that reflects both current and future business environments
  - Identify critical issues and areas of focus for business change
  - Guide decisions for use of process, information/knowledge, organization and technology
  - Ensures that any tactical decision making is aligned with the overall business goals and objectives
  - Provides a business driven framework for subsequent definition of IT Architecture

Example:

- Key business strategy provide business context for SOA target architecture:
  - Expectations of Multi-channel, device usage (Portal, Fat, Thin, Rich, Voice, Wireless, Kiosk)
  - Visibility into Business Processes and workflow
  - Customer centricity and Segmentation
  - Vision and use of Client Information File (CIF)
  - Enterprise Document Management
  - Data management needs & how data are to be shared
  - Reusability vision of business and technology services
  - Business Rules Engine as a service
A Business Architecture bridges the Component Based Model to Technical Audiences

Business Architecture provides a bridge from Strategy to Technology Vision

<table>
<thead>
<tr>
<th>Channels</th>
<th>Assisted Connected</th>
<th>Assisted Mobile</th>
<th>Self-serve</th>
</tr>
</thead>
<tbody>
<tr>
<td>Employees</td>
<td>Personal Bankers</td>
<td>Personal Financial Service Rep</td>
<td>Mobile Employees</td>
</tr>
<tr>
<td></td>
<td>Rich Client</td>
<td>Thin Client</td>
<td>Mobile Employees</td>
</tr>
</tbody>
</table>

User Presentation Services

User Interaction and Business Services

Sales
- Sales Administration
- Client Acceptance
- Needs Assessment
- Solution Presentation

Customer Management
- Client Analysis
- Client Profile
- Client Communications

Fulfilment & Maintenance
- Bank Decisioning
- Financials
- Client Arrangement
- Information Validation
- Collections & Recovery
- Handling Document

Financials & Portfolio Management
- Treasury
- General Ledger & Account Reconciliation

Business Support Services and Enterprise Systems Services

Infrastructure Services

Business Component Model provides the Business Strategy
Business services can also be described in the context of a specific Business scenario and bridge into a conceptual solutions architecture.
Business Architecture describes the business strategy for a given core competency, and provide the context for further Technical Architecture Decomposition.

1. Customers
   - Retail Consumer
   - Private Client
   - Small Business
   - Mid Market
   - Large Cap
   - Internal customers

2. Needs
   - Moving
   - Retiring
   - Starting a Business
   - Shopping for Cars
   - New Baby
   - Getting Married
   - New Job
   - Kids off to University

3. CORE CAPABILITIES
   - Lend Money
   - Move Money
   - Borrow Money
   - Manage Money
   - Exchange Money
   - Protect Assets
   - Invest Money
   - Services

4. Channels
   - Internet
   - Sales force
   - Wireless / Mobile / RIM
   - Branch
   - ATM
   - Mail
   - Telephone

5. Organization
   - LOB

Organizational structure is based on roles, responsibilities

Process capabilities provide the ability for an institution to possess core competencies within Lending Money, borrowing Money etc.

Institution Facing:
- Institution must develop “Core Competencies” which are in demand by customers
- Institution requires an Organizational structure to manage and control these competencies
- Institution requires internal process capabilities that contribute to building and delivery against these “Core competencies”

Customer Facing:
- There are multiple customer segments
- Customers have varying needs
- Customers may enter any delivery channel
- Customers expect servicing through their requested delivery channel
Agenda

- The Business Motivation of SOA
- Best Practices for Operationalizing SOA Business Transformation
  1. Take on a Program Approach for your SOA Implementation
  2. Use a Business Architecture to Provide Guidance for Technical Solution Designs
  3. Build a Prescriptive SOA Transformation Technology Architecture
  4. Create Pragmatic Roadmap and Demonstrate Continuous Progress
  5. Cross Project Service Reuse planning
  6. Pay Attention to implementing products and tools for Common Services
  7. Special Care with Coexistence and Migration for Existing Applications
A Program Level Target Architecture provides a blueprint to govern successful incremental, consistent SOA Project Implementation.
The On Demand Operating Environment (ODOE) provides a framework for the technology architecture.
Mortgage Architecture Overview

Application Services

User Interaction Services
- Collaboration
- Client Profile
- Sales Session
- Credit Application
- Lawyer Portal

Process Management Services
- Mortgage
- Open
- Renew
- Refinance
- Transaction

Business Function Services
- Client Profile
- Score
- Credit Application
- CRUD Services

Business Support Services
- Reporting (management)
- External
- Email
- Printing
- Content Manager
- Document Assembly

Enterprise Information Systems
- Loan application
- IBMS
- SRF
- CRM
- PVR
- URF

Integration Services
- Dataflows
- Business Connections
- Web Services
- Application Connectivity Services
- JCA, JDBC, JMS
- Enterprise Service Bus
- Events
- TCP/IP/SNA

Utility Services
- Rating
- Peering
- Settlement

Service Level Automation and Orchestrations
- Problem Management
- Security Services
- Workload Services
- Configuration Services
- Availability Services
- Anti-Intrusion
- Non-Repudiation
- Firewalls
- User Management

Security Services
- Authentication
- Authorization
- Encryption
- Network
- Resource Mapping

Resource Virtualization Services
- Server
- Storage
- Network

Middleware Services
- Transaction Management
- Business Activity Monitoring
- Logging
- Auditing
- Caching
- Session Management
- Application Alerts

IBM Global Services
© Copyright IBM Corporation 2005
A Prescriptive Technical Architecture Model and Templates are key for Development Consistency

- **Solution Architecture Models must be prescriptive and implementable**
  - Include Application and Infrastructure levels
  - Technical Architecture need to be specify how to apply the model and technology
  - Include guidance on how to support the functional needs of projects in the enterprise
  - Consistent use of object modeling
  - Maximize reusability across all layers
  - Provides consistent application design templates that supports the business in a cost effective, operationally efficient way

- **Example:**
  - Template for Multi-channel Access Application
  - Template for Workflow and Process-Oriented Application
SOA Operationalization require Architect and Technical Teams to possess new skills

- The SOA Architecture team at the program level require different skill sets
  - Must be able to understand the broader business picture, the different SOA layer, and how they interact

- Enterprise Architecture and Design authority must be continuously involved in the SOA Architecture design and implementation

- Need to have Transformational Architecture competencies
  - Implementation Experience is important
  - Pragmatism over ‘purist’
  - Need people that has done it before
  - Require depth in particular technologies
  - Understanding of reference architecture and patterns
  - Business Domain Knowledge are important as SOA requires much more business context
  - Need more Systems Integration Experience and skills
  - Understanding of Operational and Non-functional Requirements

- Architects require consulting, presentation & selling skills
  - Sustaining transformation program require conscious and continuous selling to the business
Agenda

- The Business Motivation of SOA

- Best Practices for Operationalizing SOA Business Transformation
  1. Take on a Program Approach for your SOA Implementation
  2. Understand your Business Strategy
  3. Build a SOA Transformation Technology Architecture
  4. Create Pragmatic Roadmap and Demonstrate Continuous Progress
  5. Cross Project Service Reuse planning
  6. Pay Attention to implementing products and tools for Common Services
  7. Special Care with Coexistence and Migration for Existing Applications
Putting it all together—SOA Implementation Phasing

Implementation Phasing Development Thought Process

This analysis identifies current states and desired target states based on business goals. Incremental roadmaps are developed to achieve these states.

Customer Business Goals and Imperatives

Scope and Comprehensiveness of Services

- **Assess current state**
- **Determine future state**
- **Identify required capabilities and initiatives**
- **Develop Phasing Roadmap**

Domain of Capability

- Business Function Services
- Information Services
- Common IT Services
- Infrastructure Services

Cross-domain enablers
### Business Focused SOA Implementation Roadmap Considerations

| There is no “pure SOA” green field installation | Must take into consideration business and technology infrastructure already in place  
<table>
<thead>
<tr>
<th></th>
<th>Coexistence &amp; Migration costs and efforts should not be overlooked</th>
</tr>
</thead>
</table>
| Pragmatic vs. “boiling the ocean”              | Do simple scenario first and add-on complexity vs. most complex scenario  
|                                               | Foundational components must be done SOA Infrastructure need to be added incrementally  
|                                               | Adding or dropping functions/projects must understand inter-dependencies |
| Project Status & Expectation Management       | Keep track of project status and continue to fine-tune expectations  
|                                               | Establish key milestones & expected outcome, entrance and exit criteria |
| Synchronize with other Enterprise Initiatives | Roadmap must be synchronized with other sustaining business initiatives |
| Cross Reference Services for each project     | Cross referencing service components needed for each project, inter-dependency of projects need constant validations  
|                                               | - Be careful when spreading cost and value of common services components across projects – may lose estimated efforts  
|                                               | - Pure SOA infrastructure Service interspersed with Transformation projects that can bring value |
Agenda

- The Business Motivation of SOA

- Best Practices for Operationalizing SOA Business Transformation
  1. Take on a Program Approach for your SOA Implementation
  2. Understand your Business Strategy
  3. Build a SOA Transformation Technology Architecture
  4. Create Pragmatic Roadmap and Demonstrate Continuous Progress
  5. **Cross Project Service Reuse planning**
  6. Pay Attention to implementing products and tools for Common Services
  7. Special Care with Coexistence and migration for non-interrupted business
Reusing Services for multiple projects require thorough planning and decision process

- Need to analyse behavioural requirements of the candidate application contexts at the same level of detail.
  - Business Processes (e.g. Loan Origination, Request to quote)
  - Business Services (e.g. Pricing engine, Product configurator)
  - Embedded Services (e.g. check price, check availability)

- **Bottom up approach**
  - Faster initial deployment
  - Fine-grained services & code requires thorough planning
    - What it will be used for
    - How it is to be done
    - Difficult to instill flexibility in the wrapper for all situations

- **Top down approach**
  - Must combine with an overall architecture to guide the decision
  - Start with coarse grained business processes
  - Will provide higher probability of reuse

- Systems Management and Security of Service components also critical
  - How to monitor the services?
  - How to roll back and recover when business process if spread over multiple services?
  - If monitored at too low a level, can be very expensive

---

**Examples**

<table>
<thead>
<tr>
<th>How many?</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tens of processes</td>
<td>Request to quote, Loan origination, Product development</td>
</tr>
<tr>
<td>Hundreds of services</td>
<td>Product configurator, Pricing engine, Inventory status</td>
</tr>
<tr>
<td>Thousands of service installations</td>
<td>Configure product, Check price, Check availability</td>
</tr>
</tbody>
</table>

---

Source: Organic Business, Forrester Research
Fast Track to Design Reusable Process Components
In order to develop potentially reusable services, a business application project should perform the following process:

1. Identify Services (employ SOMA analysis techniques)
2. Categorize Services
3. Identify Reusable Services (via Decision Model)
4. Identify Appropriate Build Strategy for Reusable Services (via Decision Model)
5. Develop Reusable Services as per build strategies.
6. Integrate Services
7. Test and Deploy

- Accurate service identification truly begins in the Analysis and Design phase
- Each reusable Service should have its own Transformation Strategy
Buy vs. Build vs. Refactoring of Service Components

- Pre-requisite : Avoid Jumping into Coding!!
- Factors for driving Buy / Build/ Refactor Decisions
  - Total Cost of Ownership
    - Design and Development Costs
      - Application Architecture
      - Application Functionality
    - Integration Costs
    - Training
    - Maintenance
    - Systems Management & Operations costs
    - License
    - Opportunity Costs
  - Amount of Reuse
    - Enterprise / Departmental / Project level of Reuse
- A Hybrid approach will probably be the most likely scenario
## Services Reuse and Transition Considerations

| The Services defined need to have its own transition strategy | • What is the lowest functional services that can be created and used by the various business components  
• What is the next phase services (enhanced functionality) and when is the best time to deploy?  
• How to deploy these services without impacting production operations of multiple processes?  
• How does a business process decide when or how to adopt? |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Do not only consider happy path</td>
<td>• Too much exception process may also kill your reusability scheme</td>
</tr>
</tbody>
</table>
| One time deployment on a single application is simple | • Multiple processes and multiple applications will need attention to governance, technical specifications, and full end to end considerations.  
• Need to consider how services design can accommodate changing business requirement  
• Need to consider traceability of all services  
• Once identified all services, need to have a way to group, regroup and re-aggregate to be used in multiple projects |
| Testing & Performance Consideration | • Performance, system testing and regression testing implications need to be understood |
Agenda

- The Business Motivation of SOA

- Best Practices for Operationalizing SOA Business Transformation
  1. Take on a Program Approach for your SOA Implementation
  2. Understand your Business Strategy
  3. Build a SOA Transformation Technology Architecture
  4. Create Pragmatic Roadmap and Demonstrate Continuous Progress
  5. Cross Project Service Reuse planning
  6. Pay Attention to implementing products and tools for Common Services
  7. Special Care with Coexistence and migration for non-interrupted business
Process Choreography is one of the most powerful concepts to support an SOA Composite Application

- Process choreography essentially knits together processes executing in an SOA infrastructure, such as process execution engines and applications that encapsulate process, to provide an end-to-end view of a business process.

**Business Process**
- Long running
- One or more persons interacting
- Multiple valid business process states
- Alternative workflows for non-normal conditions

**Business Transaction**
- Short term, non-interactive
- One change of business state
- Consumes one or more enterprise service
- Targeted level of service reuse
- Loose coupling important
- May require compensation

**Function Service**
- Collaborations to implement a single web service
- Collaborating apps encapsulated via web services
- Performance favored over loose coupling
### Business Process Management Implementation for SOA Considerations

| **Business Process Management and BPMS are not the same** | • Decide whether you need business performance monitoring, or workflow  
• BPM enables easier transition and workflow but not a silver bullet  
• BPMS provides the Business monitoring and simulation functions for business |
| --- | --- |
| **Some BPM tools require more custom coding** | • May force high level of custom code and very expensive  
• Not just a workflow manager  
• Require careful analysis of legacy application if plan to use in parallel |
| **Do not use BPM as Integration hub** | • Should not custom built Integration code  
• Use Standard Integration Software, out-of-the-box tooling and standard API  
• open standards preferred |
| **Adopt a progressive workflow strategy** | • Adapt a progressive workflow strategy  
  - Only automate what you must. Keeping something manual is OK  
  - Do not try to choreograph to too fine-grain (unless you understand how you want to manage it)  
• Use of ISV packages does not make BPM work easier |
| **Understand where is the control between the new flow and the old application flow** | • Synchronization problems caused  
  - Forward State Management to BPM is easier  
  - Reverse State Management (to update legacy systems) is very difficult and may cause data integrity issues  
• Make sure you BPM provides compensation transaction function |
| **Do lots of prototypes before committing designs** | • Development concept very different than traditional application development |
A Proven Successful Workflow Strategy takes a route of end to end management control before turning to progressive complexity

Recommendation:
• Light-weight workflow deployment and progressive STP (Straight Through Processing) to maximize benefit realization, process efficiency and reusability
  • Rationalization of cross product business process rather than specific product processes (e.g. Open Process)
  • Consistent use of utility processes (e.g. Document Capture process) for cross product process reuse
  • Rules and Business Logic to be associated with process components
  • Business Benefits analysis before jumping to automation

Process Re-design / Simplification / Reusability Design, Business Modelling

Refine and Extend Key Processes with defined Benefits, Service Oriented Processes

Maximize Automation

Intermediate Automation

Progressive STP

Workflow Lite

Progressive Selection of Straight Through Processes, Optimize Flow

Full STP
## Rules Engine Implementation for SOA Considerations

<table>
<thead>
<tr>
<th>Rules are mostly embedded in code</th>
<th>• Not always feasible to externalize rules</th>
</tr>
</thead>
</table>
| Business Rules usage and deployment strategy central to SOA transformation | • May need different rules strategy and tool for different rule types  
• Require careful categorization, Maintenance strategy |
| Some rules engine have BPM workflow engines attached to it (and vice versa) | • Competing workflow  
• May have several specialized rules engine (e.g. Product Configurator Rules Engine, Rating Rules Engine, Decision Engines) |
| Performance considerations for using Rules Engines | • Each process can involve lots of trips to the rules engine  
• May cause too many hops and multiple points of failure |
Agenda

- The Business Motivation of SOA

- Best Practices for Operationalizing SOA Business Transformation
  1. Take on a Program Approach for your SOA Implementation
  2. Understand your Business Strategy
  3. Build a SOA Transformation Technology Architecture
  4. Create Pragmatic Roadmap and Demonstrate Continuous Progress
  5. Cross Project Service Reuse planning
  6. Pay Attention to implementing products and tools for Common Services
  7. Special Care with Coexistence and Migration of Existing Applications
Incremental Transformation of Monolithic Applications require extreme care

- Breaking up an Existing code is harder than expected
- Legacy code generally not modularized
  - Including Client/Server and 1st Generation OO technologies
- Big Bang Conversion is not possible
- Incremental Transformation require old and new systems to co-exist and work together for a long time
  - Navigation between old screens and new screens within a process difficult (e.g. in case of Account Open which is part of highly intertwined processes)
  - Keeping state between old and new systems is important but has to be selective
  - Extracting single process out of old app. is very difficult
- Do not rush out to get a tool
  - There is no perfect code generators
  - converting bad code to a new bad code is not recommended
  - Use of a Business Process Engine and a new UI may help
  - Use of a tool to help with Application Mining is also helpful
    - Have to do focused mining
    - Do not expect the tool to do it all
Summary

Best Practices for Operationalizing SOA Business Transformation

1. Take on a Program Approach for your SOA Implementation
2. Understand your Business Strategy
3. Build a SOA Transformation Technology Architecture
4. Create Pragmatic Roadmap and Demonstrate Continuous Progress
5. Cross Project Service Reuse planning
6. Pay Attention to implementing products and tools for Common Services
7. Special Care with Coexistence and migration for non-interrupted business
Thank You!